

AMENDMENTSIn the Claims

1. **(Currently Amended)** A method comprising:  
providing a debugger agent, wherein  
the debugger agent is configured to select a debugger program suitable for a  
[[unit]] device under test, wherein and  
the [[unit]] device under test includes is configured to execute a program  
under test;  
causing the debugger agent to [[be]] load[[ed]] the debugger program into the [[unit]]  
device under test;  
sending a plurality of test commands to the [[unit]] device under test according to a test  
script; and  
activating the debugger program when a watched event occurs during execution of the  
program under test.
2. **(Currently Amended)** The method of claim 1 further comprising:  
directing a debugger command to the debugger program; and  
recording information provided by the debugger program according to the debugger  
command.
3. **(Currently Amended)** The method of claim 1 further comprising:  
pausing execution of the program under test; and  
allowing a user to control the debugger program.
4. **(Currently Amended)** The method of claim 1 further comprising:  
invoking the debugger program while specifying the program under test as a target of  
the debugger program.
5. **(Currently Amended)** The method of claim 1 further comprising:

instructing the debugger program to associate itself with a process executing on the [[unit]] device under test, wherein the process corresponds to the program under test.

6. **(Currently Amended)** The method of claim 1 further comprising:  
sending a command to the debugger program, wherein the command performs at least one of :

setting a breakpoint in the program under test;  
setting a watchpoint in the program under test;  
setting a catchpoint in the program under test; and  
setting a tracepoint in the program under test[[;]].

7. **(Currently Amended)** The method of claim 1 wherein the watched event includes comprises at least one of:

a processor exception[[,]];  
a program under test error[[,]];  
reaching a breakpoint in the program under test;  
reaching a watchpoint in the program under test;  
reaching a catchpoint in the program under test; and  
reaching a tracepoint in the program under test.

8. **(Currently Amended)** The method of claim 1 further comprising:  
selecting a platform-specific debugger program corresponding to a processor in the [[unit]] device under test; and  
loading the platform-specific debugger program into the [[unit]] device under test.

9. **(Currently Amended)** The method of claim 8 further comprising:  
loading, into the [[unit]] device under test, a symbol file corresponding to the program under test.

10. **(Currently Amended)** A system comprising:  
a memory;

a processor coupled to the memory; and  
a debugger agent, wherein at least a portion of the debugger agent is encoded as  
instructions stored in the memory and executable on the processor, and wherein  
the debugger agent is configured to:  
select a debugger program suitable for a [[unit]] device under test, wherein  
the [[unit]] device under test includes is configured to execute a program  
under test;  
cause the debugger program to be loaded into the [[unit]] device under test;  
send a plurality of test commands to the [[unit]] device under test according to a  
test script; and  
activate the debugger program when a watched event occurs during execution of  
the program under test.

11. (Original) The system of claim 10 further comprising at least one debugger program stored in at least one of the memory and a storage device accessible by the processor.
12. (Original) The system of claim 10 further comprising at least one symbol file stored in at least one of the memory and a storage device accessible by the processor.
13. (Original) The system of claim 10 further comprising:  
a test script handler, wherein at least a portion of the test script handler is encoded as  
instructions stored in the memory and executable on the processor.
14. (Original) The system of claim 13 wherein the test script handler is further configured to send the plurality of test commands to the debugger agent.
15. (Original) The system of claim 10 further comprising:  
a second memory;  
a second processor coupled to the second memory; and  
a test script handler, wherein at least a portion of the test script handler is encoded as  
instructions stored in the second memory and executable on the second processor.

16. (Original) The system of claim 15 wherein the test script handler is further configured to send the plurality of test commands to the debugger agent.
17. (Original) The system of claim 10 wherein the debugger agent is further configured to:  
direct a debugger program command to the debugger program; and  
record information provided by the debugger program according to the debugger command.
18. (Original) The system of claim 10 wherein the debugger agent is further configured to:  
suspend execution of the program under test; and  
allow a user to control the debugger program.
19. (Original) The system of claim 10 wherein the debugger agent is further configured to:  
invoke the debugger program while specifying the program under test as a target of the debugger program.
20. (**Currently Amended**) The system of claim 10 wherein the debugger agent is further configured to:  
command the debugger program to associate itself with a process executing on the [[unit]] device under test, wherein the process corresponds to the program under test.
21. (**Currently Amended**) The system of claim 10 wherein the debugger agent is further configured to:  
send a command to the debugger program, wherein the command performs at least one of:  
setting a breakpoint in the program under test;  
setting a watchpoint in the program under test;  
setting a catchpoint in the program under test; and

setting a tracepoint in the program under test[[;]].

22. (Currently Amended) The system of claim 10 wherein the watched event includes comprises at least one of a processor exception, a program under test error, reaching a breakpoint in the program under test[[;]], reaching a watchpoint in the program under test[[;]], reaching a catchpoint in the program under test[[;]], and reaching a tracepoint in the program under test.

23. (Currently Amended) The system of claim 10 wherein the debugger agent is further configured to:

select a platform-specific debugger program corresponding to a processor in the [[unit]] device under test; and

load the platform-specific debugger program into the [[unit]] device under test.

24. (Currently Amended) The system of claim 23 wherein the debugger agent is further configured to:

load, into the [[unit]] device under test, a symbol file corresponding to the program under test.

25. (Currently Amended) A computer readable storage medium comprising program instructions executable on a processor, the computer readable storage medium ~~being at least one of an electronic storage medium, a magnetic storage medium, an optical storage medium, and a communications medium conveying signals encoding the program~~ instructions, wherein the program instructions are operable configured to implement each of:

providing a debugger agent, wherein

the debugger agent is configured to select a debugger program suitable for a

[[unit]] device under test, wherein and

the [[unit]] device under test includes is configured to execute a program under test;

causing the debugger agent to [[be]] load[[ed]] the debugger program into the [[unit]] device under test;

sending a plurality of test commands to the [[unit]] device under test according to a test script; and

activating the debugger program when a watched event occurs during execution of the program under test.

26. (Currently Amended) The computer readable storage medium of claim 25 further comprising program instructions operable to implement each of:  
directing a debugger command to the debugger program; and  
recording information provided by the debugger program according to the debugger command.

27. (Currently Amended) The computer readable storage medium of claim 25 further comprising program instructions operable to implement each of:  
pausing execution of the program under test; and  
allowing a user to control the debugger program.

28. (Currently Amended) The computer readable storage medium of claim 25 further comprising program instructions operable to implement:  
invoking the debugger program while specifying the program under test as a target of the debugger program.

29. (Currently Amended) The computer readable storage medium of claim 25 further comprising program instructions operable to implement:  
instructing the debugger program to associate itself with a process executing on the [[unit]] device under test, wherein the process corresponds to the program under test.

30. (Currently Amended) The computer readable storage medium of claim 25 further comprising program instructions operable to implement:  
sending a command to the debugger program, wherein the command performs at least one of:  
setting a breakpoint in the program under test;

setting a watchpoint in the program under test;  
setting a catchpoint in the program under test; and  
setting a tracepoint in the program under test[[;]].

31. **(Currently Amended)** The computer readable storage medium of claim 25 wherein the watched event includes comprises at least one of a processor exception, a program under test error, reaching a breakpoint in the program under test[[;]], reaching a watchpoint in the program under test[[;]], reaching a catchpoint in the program under test[[;]], and reaching a tracepoint in the program under test.

32. **(Currently Amended)** The computer readable storage medium of claim 25 further comprising program instructions operable to implement each of selecting a platform-specific debugger program corresponding to a processor in the [[unit]] device under test; and loading the platform-specific debugger program into the [[unit]] device under test.

33. **(Currently Amended)** The computer readable storage medium of claim 25 further comprising program instructions operable to implement: loading, into the [[unit]] device under test, a symbol file corresponding to the program under test.

34. **(Currently Amended)** An apparatus comprising:  
a means for causing a means for debugging a program under test to be loaded into a [[unit]] device under test, wherein the [[unit]] device under test includes is configured to execute the program under test;  
a means for sending a plurality of test commands to the [[unit]] device under test according to a test script; and  
a means for activating the means for debugging when a watched event occurs during execution of the program under test.

35. **(Original)** The apparatus of claim 34 further comprising:  
a means for directing an instruction to the means for debugging a program under test; and

a means for recording information provided by the means for debugging a program under test.

36. (Original) The apparatus of claim 34 further comprising:  
a means for pausing execution of the program under test: and  
a means for allowing a user to control the means for debugging a program under test.

37. (**Currently Amended**) The apparatus of claim 34 further comprising:  
a means for instructing the means for debugging a program under test to associate itself with a process executing on the [[unit]] device under test, wherein the process corresponds to the program under test.

38. (**Currently Amended**) The apparatus of claim 34 further comprising:  
a means for sending a command to the means for debugging a program under test,  
wherein the command performs at least one of :  
setting a breakpoint in the program under test;  
setting a watchpoint in the program under test;  
setting a catchpoint in the program under test; and  
setting a tracepoint in the program under test~~[[;]]~~.

39. (**Currently Amended**) The apparatus of claim 34 wherein the watched event includes comprises at least one of a processor exception, a program under test error, reaching a breakpoint in the program under test~~[[;]]~~, reaching a watchpoint in the program under test~~[[;]]~~, reaching a catchpoint in the program under test~~[[;]]~~, and reaching a tracepoint in the program under test.